DeNC: Unleash Neural Codecs in Video Streaming with Diffusion Enhancement

Qihua Zhou[1], Ruibin Li[2], Jingcai Guo[2], Yaodong Huang[1], Zhenda Xu[2], Laizhong Cui[1]*, Song Guo[3]

[1]College of Computer Science and Software Engineering, Shenzhen University
[2]Department of Computing, The Hong Kong Polytechnic University
[3]Department of Computer Science and Engineering, The Hong Kong University of Science and Technology
{qihuazhou, yd.huang, cuilz}@szu.edu.cn
{ruibin.li, jackal.xu}@connect.polyu.hk, jc-jingcai.guo@polyu.edu.hk,
songguo@cse.ust.hk

Abstract

Recent years have witnessed the rise of *Neural-enhanced Video Streaming* (NeVS), which integrates neural restoration models into video codecs for higher compression-restoration performance. Despite its benefit, existing work has not well explored the full potential of NeVS paradigm, due to: (1) post-streaming restoration by decoder while lacking the proactive collaboration of encoder, (2) end-to-end optimization based on conventional rate-distortion theory, which has been verified that low distortion is not always a synonym for high perceptual quality, and (3) coupled design for domain-specific tasks that cannot generalize to various video codecs. Observing these limitations, our objective is not to incrementally present an improved restoration model. Instead, we focus on the encoder-decoder synergy, *i.e.,* the codec, which is non-trivial since it inherently strikes the *rate-distortion-perception* trade-off of NeVS. Aiming at this target, we propose the *Diffusion-enhanced Neural Codec* (DeNC), a plug-and-play module for current NeVS paradigm, to significantly reduce the required bitrates while preserving high perceptual quality of restored videos. Our key design is twofold. First, DeNC improves the encoder's compression efficiency by simultaneously reducing the resolution and color bit-depth of frame referencing. Second, DeNC empowers the decoder with perception-oriented restoration capability by making its diffusion-based restoration process aware of the encoder's compression conditions. Real-world evaluations show that DeNC improves compression ratios with nearly an order of magnitude and achieves much higher restoration quality (*e.g.,* 93+ VMAF and 23% higher MOS) over the latest baselines.

Introduction

Video streaming has been a fundamental infrastructure for today's Internet services, *e.g.,* YouTube, Netflix and Zoom. Generally, video streaming involves two sides: (1) *ingestion* where the streamer uploads video bitstreams to the media server through streamer's uplink (Li, Li, and Lu 2021), and (2) *distribution* where the server delivers transcoded videos to viewers through its downlink (Liu et al. 2020). Since the video ingestion quality relies on streamer's uplink condition, the downstream viewer's experience will directly suffer when ingestion side is congested (Yang et al. 2022; Narayanan et al. 2021).

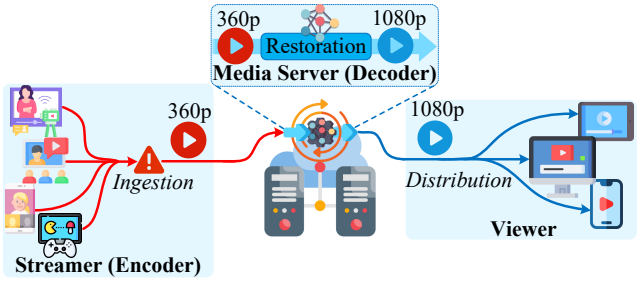
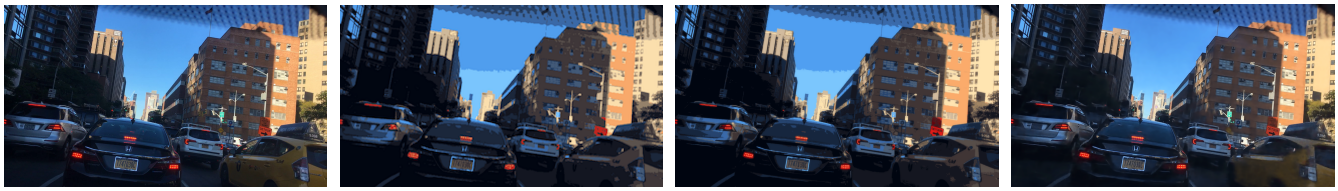


Figure 1: The overview of NeVS's compression-restoration pipeline (Yeo et al. 2022; Kim et al. 2020; Yeo et al. 2020).

Recently, the *Neural-enhanced Video Steaming* (NeVS) paradigm has shown great promise to alleviate this issue (Du et al. 2022; Dasari et al. 2022). As shown in Figure 1, raw videos are compressed in low quality for streaming and then restored by the media server via an integrated neural restoration model (Zhang et al. 2022). Specifically, the compression can be handled at frame level in two aspects: (1) downscaling the resolution, *e.g.,* from 1080p to 360p (Yeo et al. 2022; Kim et al. 2020) and (2) reducing the color bit-depth of pixel representation, *e.g.,* from 24-bit to 4-bit (Liu et al. 2022). In NeVS's pipeline, a neural network, instead of hand-crafted logic is utilized for frame encoding and decoding, so as to achieve higher compression-restoration performance. We call this *neural codec* (Cheng et al. 2024; Chen et al. 2024). By using neural codecs, the core objective of NeVS is to save the required bitrates while preserving high perceptual quality of restored videos, *i.e.,* achieving a high *rate-distortion-perception* trade-off (Wang et al. 2022).

However, preliminary experiments in Figure 2 have verified that current neural codecs may not always maintain this trade-off, especially when videos are highly compressed in resolution and color bit-depth. We can observe that the perceptual quality of restored video is significantly degraded when the original video is compressed with low resolution and color bit-depth, even with the state-of-the-art (SOTA) Gemino (Sivaraman et al. 2024) codec. We reveal that existing neural codecs have not well explored NeVS's potential, due to: (1) relying on post-streaming restoration on decoder

*Corresponding author.



(a) Original 24-bit 1080p. (b) Compressed 4-bit 360p. (c) Gemino restored 24-bit 1080p. (d) Our restored 24-bit 1080p.

Figure 2: Visualization of a frame in the NeVS pipeline: (a) original 24-bit 1080p, (b) 4-bit 360p by encoder's resolution-color compression, (c) restored 24-bit 1080p by Gemino's decoder, and (d) restored 24-bit 1080p by our DeNC's decoder with much higher perceptual quality, *e.g.,* see the sky color transition and car license number. **Best viewed in color and zoomed-in**.

while lacking proactive collaboration of encoder (Yeo et al. 2022; Kim et al. 2020), (2) optimizing the compression-restoration pipeline based on conventional rate-distortion theory, which has been verified that low distortion is not always a synonym for high perceptual quality (Cheng et al. 2024; Hu, Lu, and Xu 2021), and (3) coupling the neural restoration design with domain-specific tasks, thus cannot generalize to different video codecs (Chen et al. 2024; Sivaraman et al. 2024). These observations motivate us to conduct the encoder-decoder (*i.e.,* codec) synergy instead of incrementally presenting an improved restoration model.

Aiming at this target, we propose the *Diffusion-enhanced Neural Codec* (DeNC), which efficiently restores the low-quality videos by leveraging the perception-oriented generative property of diffusion models. The key design is twofold. First, DeNC improves the encoder's compression efficiency by simultaneously reducing resolution and color bit-depth of video frames. Second, DeNC provides the decoder with perceptual quality enhancement by making the diffusion restoration process aware of encoder's resolution-color compression. This codec-level design philosophy makes DeNC compatible with existing video codecs, *e.g.,* H.265 (H.265 2025), VP9 (Google 2025) and AV1 (AOMedia 2025), and the benefits of conventional codecs are still well preserved. Thus, DeNC can serve as a plug-and-play module to upgrade current NeVS paradigm. Extensive experiments on public cloud services with YouTube videos show that DeNC significantly improves compression ratios with nearly an order of magnitude and achieves much higher restoration quality, *e.g.,* 23% higher mean opinion score (MOS), over the latest methods. Overall, our key contributions are as follows.

- **Unleash the potential of NeVS.** To the best of our knowledge, DeNC is the first work to inherently improve NeVS's *rate-distortion-perception* trade-off, which extends the usage boundary of the NeVS paradigm.
- **In-depth encoder-decoder synergy.** DeNC effectively improves the encoder's compression ratio and reduces the video delivery bitrates by an order of magnitude. It explores the perceptual-generative property of diffusion models to refine decoder's restoration capability, holding strong robustness in different quality assessment metrics.
- **General design for various video codecs.** DeNC is a general compression-restoration pipeline and is compatible with existing video codecs. It can serve as a plug-and-play module to upgrade current NeVS methods.

Related Work

Neural-enhanced video streaming. Recall the video streaming infrastructure in Figure 1, alleviating the uplink bottleneck is a crucial issue on the ingestion side, which promotes the rise of *Neural-enhanced Video Streaming* (NeVS) (Yeo et al. 2022; Liu et al. 2021; Du et al. 2022; Kim et al. 2020; Yeo et al. 2020; Dasari et al. 2022) paradigm. To improve the video delivery performance, NeVS involves the collaboration between streamer and server. First, the streamer compressed the original high-quality video frames into the low-quality version, then encodes the frames into video bitstreams for network transmission. The media server decodes the bitstreams as a series of frames and fed them into a neural restoration model for quality enhancement (Yeo et al. 2022; Zhang et al. 2022; Kim et al. 2020; Yeo et al. 2020). The final restored video holds comparable visual experience as the original version, thus can be applied to the content distribution in different downstream tasks. Recently, optimizing the NeVS pipeline has become a hot topic, including improving video encoding efficiency (Du et al. 2022; Dasari et al. 2022), reducing steaming latency (Yeo et al. 2018), designing adaptive bit-rate delivery (Dasari et al. 2022) and optimizing frame referencing (Yeo et al. 2022). Overall, the core objective of the NeVS paradigm is to improve the overall *rate-distortion-perception* trade-off, *i.e.,* reducing streaming bitrates while maximizing perceptual quality after restoration (Wang et al. 2022).

Video neural codecs and restoration models. Conventional video codecs, *e.g.,* H.265 (H.265 2025), VP9 (Google 2025) and AV1 (AOMedia 2025), usually compress video bitstreams by removing redundancy through intra/inter-frame prediction, motion estimation and compensation. However, these techniques fall short in insufficient compression ratios and and further compression can easily degrade video quality. Fortunately, the hand-crafted components in conventional codecs can be replaced by neural networks, which learn spatial-temporal patterns and hold higher restoration capability at the same streaming bitrate. Therefore, neural codec is a key component of NeVS paradigm. As to the integrated neural restoration models, video super-resolution (SR) is a natural choice, which transfers the low-resolution frames to high-resolution ones, *e.g.,* IconVSR (Chan et al. 2021), RealBasicVSR (Chan et al. 2022b) and BasicVSR++ (Chan et al. 2022a). Some advanced restoration techniques leverage the generative networks to com-

pensate frame spatial-temporal coherence, *e.g.,* TecoGAN (Chu et al. 2020), PULSE (Menon et al. 2020) and Real-ESRGAN (Wang et al. 2021). Recently, the diffusion probabilistic models (*e.g.,* DDPM (Ho, Jain, and Abbeel 2020), DDIM (Song, Meng, and Ermon 2021), LDM (Rombach et al. 2022) and SR3 (Saharia et al. 2023)) have achieved impressive performance in diverse computer vision tasks, including inpainting (Lugmayr et al. 2022), colorization (Song et al. 2021) and image synthesis (Saharia et al. 2023). The readers can refer to following survey papers to learn more about NeVS (Shi et al. 2024), neural codecs (Lee, Venieris, and Lane 2022; Xu et al. 2024) and diffusion models (Cao et al. 2024). Here, our insight is to utilize the inherent generative property of diffusion process for higher *rate-distortion-perception* trade-off, which motivates us to propose the *Diffusion-enhanced Neural Codec* (DeNC) to upgrade current NeVS paradigm.

Diffusion-enhanced Video Codec

Problem Formulation and Objective

Traditional video streaming. As a video consists of a series of frames $\mathbf{x}$, we use x_i to denote an original raw high-quality frame with index i, where $x_i \in \mathbf{x}$ and i identifies the sequence order for video encoding. After encoding all the frames as a high-quality video, the video bitstreams will be uploaded through the network to the media server. The server receives the bitstreams and decodes it for downstream tasks. We can adopt common video standards, *e.g.,* H.265 (H.265 2025), VP9 (Google 2025) and AV1 (AOMedia 2025), to handle the entire encoder-decoder procedure and formulate the traditional video streaming pipeline $f(\cdot)$ as: $f(\mathbf{x}) = \texttt{Decode}(\texttt{Encode}(\mathbf{x}))$.

Neural-enhanced video streaming. Recent NeVS research (Yeo et al. 2022; Du et al. 2022; Kim et al. 2020; Yeo et al. 2020; Dasari et al. 2022) has shown that directly encoding the video from raw frames and delivering the high-quality video through network is impractical due to streamer's limited uplink bandwidth. Conducting frame compression before video encoding is necessary to fit the bandwidth restriction, where reducing the frame resolution with a constant downscaling factor (*e.g.,* from 720p to 360p with a $2\times$ factor) is widely used (Yeo et al. 2022; Zhang et al. 2022; Kim et al. 2020; Yeo et al. 2020; Netflix 2025a). As a result, current NeVS paradigm usually encodes the raw videos in low quality and then delivers the video bitstreams through the network. Since the videos are compressed, a pre-trained neural network (NN) is adopted by the decoder to restore the low-quality videos to the high-quality ones (Yeo et al. 2022; Zhang et al. 2022). Thus, a typical NeVS pipeline can be described as: $f(\mathbf{x}) = \texttt{Decode}(\texttt{Encode}(\mathbf{x}; k); \texttt{NN})$, where k is the quality degradation kernel by encoder's compression.

Our encoder-decoder synergy. Different from existing work, we reveal that the color bit-depth, *i.e.,* the number of bits to represent a unique pixel in visual, has not been well exploited to further reduce the frame size. By conducting the frame compression from bit-depth and resolution perspectives simultaneously, we can achieve a much higher compression ratio over the existing methods. However, as more visual information has been compressed, the inverse restoration process is more challenging. Inspired by the perceptual-oriented generative property of diffusion models, we design the *Diffusion-enhanced Neural Codec* (DeNC), a novel neural codec, to achieve higher *rate-distortion-perception* trade-off. Figure 3 illustrates the pipeline of our DeNC's method. Together with the downscaling factor s, color bit-depth n, degradation kernel k, and diffusion model parameters $\boldsymbol{\theta}$, the entire pipeline of our DeNC can be formulated as: $f(\mathbf{x}) = \texttt{Decode}(\texttt{Encode}(\mathbf{x}; s, n, k); \boldsymbol{\theta})$.

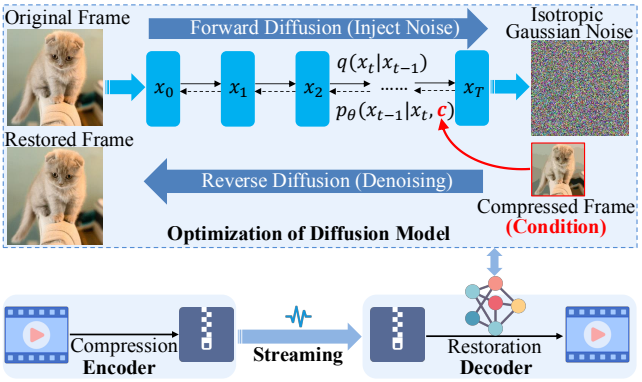


Figure 3: Pipeline of our DeNC, which restores low-quality videos from both frame resolution and color bit-depth.

Objective. By embedding the variables of s and n into diffusion conditions, our objective is to optimize DeNC by training on public video datasets to *minimize the gap between restored video frames* $f(\mathbf{x})$ *and the original ones* $\mathbf{x}$. We will discuss how DeNC handles the encoder-decoder synergy in the following sections.

Encoder with Resolution-color Compression

The role of DeNC's encoder is to reduce the video size with a much higher compression ratio over existing NeVS methods, so as to save the delivery bitrates. Note that extreme compression will lead to a significant degradation on frame visual quality, which may exceed the restoration capacity of the neural restoration model on the decoder. We cannot introduce an arbitrary compression mechanism, but need to conduct compression based on the decoder's restoration property. Thus, we handle the compression from two aspects: (1) downscaling the frame resolution and (2) reducing the pixel color bit-depth.

Patch-wise Resolution Downscaling The first compression perspective is to downscale the frame resolution, *i.e.,* shrinking the spatial size under the control of a scaling factor s. As the number of pixels within the frame is reduced in both width and height dimensions, resolution downscaling can provide a $s^2\times$ frame size reduction compared with the original frame. Although fewer pixels are used to represent a frame, its basic visual features should be preserved, otherwise, the visual quality degradation will exceed the decoder's recovery capacity. This property requires the scaling algorithm to retain the most representative pixels by analyzing the numerical distribution in each frame patch (*i.e.,*

the macroblock, a non-overlapping square block with $s \times s$ pixels in visual). As a preliminary concept to video codec, patch can serve as the basic processing units to explore intra- and inter-frame correlation (Yeo et al. 2022; Dasari et al. 2022; Kim et al. 2020; Yeo et al. 2020, 2018). In default, we suggest using 4×4 patch size, which is a sufficient fine-grained granularity to retain visual features after downscaling. Therefore, as for each pixel, we can figure out a patch where this pixel locates in the center. Specifically, considering the pixels on the frame border, we adopt zero-padding to the border with $\lceil \frac{s}{2} \rceil$ pixels in width and height, so as to guarantee complete patches. Given a scaling factor s, we can divide the frame into a series of $s \times s$ patches. Based on the patch division, we introduce a *Gaussian Blur* to the frame and smooth the features involving a junction of patches. Inside each patch, we calculate the weighted average of all the pixels inside and shrink the patch by this average. Thus, the entire frame is downscaled by $s\times$ in both width and height.

Color Bit-depth Quantization The second compression perspective is to reduce the color bit-depth, *i.e.,* the number of bits to identify a unique pixel. As to common video standards, the source frames are usually organized with 8-bit color space with RGB channels. Therefore, each pixel within a frame is represented in 24 bits, which is similar to the color space of PNG and JPEG formats (Liu et al. 2022, 2019). Meanwhile, the pixel vectors along the RBG channel hold similar distributions when they describe close colors in visual. This motivates us to revisit the *Vector Quantization* (Gray, Linder, and Gill 2008) technique and reduce the number of different colors represented by the pixels. For example, if we quantize the color space into 4 bits with 2^4 different colors, the frame size can be compressed into $4/24$ as the original frame with full-color bit-depth. The key here is to find a proper vector quantization scheme to transfer the full bit-depth pixels into low bit-depth ones. Specifically, we use the K-means clustering to handle the quantization procedure, which contains the following two steps.

Step #1: Codebook generation. This step aims at generating the quantization codebook that maps all pixels from full color bit-depth to low bit-depth, *e.g.,* from orginal 24-bit color space to 4-bit version. This requires us to group all the pixels within the frames into several clusters and represent all the pixels belonging to the same cluster by its centroid, so as to reduce the information entropy (reflected by number of bits) to identify each unique pixel. Here, the number of clusters is called the quantization level, which directly impacts the representation precision of the pixel color space. If we adopt n-bit to generate the quantization codebook, all the pixels will be grouped into 2^n clusters. In our DeNC, n is set in the range of $[4, 8]$ to greatly reduce frame size over the original 24-bit color space. Given n-bit budget to represent the color space, we choose the K-means clustering to generate the quantization codebook. Note that we need to restrict the computational overhead of calculating K-means clustering model because it iteratively calculates the neighbourhood distance for each pixel. Although we have downscaled the resolution before color quantization, the pixel number of low-resolution frame is still in order of magnitude of $10^5 - 10^6$. In this case, directly adopting K-means on the entire frame pix is computationally unacceptable. To address this challenge, we uniformly sample a subset of pixels (usually in order of magnitude of 10^3) from the frame and obtain K-means clustering model based on these samples. The K-means model describes how to map all the pixels into 2^n clusters and figures out the all cluster centroids. Each centroid is assigned with a unique index, ranging from 0 to $2^n - 1$. Therefore, the gist of our quantization codebook is to map each pixel to its corresponding cluster centroid, which can be represented by n bits.

Step #2: Pixel quantization. Based on the first step of generating quantization codebook, the second step is to replace each pixel by its cluster centroid. The original pixel matrix describing a frame can be transferred as the centroid matrix with the same shape, where each element corresponds to the pixel's centroid. As a result, the original full bit-depth pixels are quantized into n bit-depth version. The frame size is compressed to $\frac{n}{24}$ of the raw frame in 24-bit color space. In realistic deployment, the codebook can be obtained and sent to the ingestion server in advance. Therefore, the communication cost for transmitting codebook can be omitted.

Summary. Theoretically, given the scaling factor s and color bit-depth n, we can figure out the entire compression ratio over traditional 24-bit full-resolution frames as $\frac{24s^2}{n}$. Therefore, the encoder contributes to the resolution-color compression and reduces the bitrates of video streaming.

Decoder with Denoising Diffusion Restoration

On the streamer side, our DeNC's encoder aims to provide a great compression ratio to reduce video delivery bitrates. Meanwhile, we deploy DeNC's decoder on the ingestion server to recover the video bitstreams and enhance the visual quality by leveraging the visual-generative property of diffusion models. As a kind of generative model, we optimize the decoder based on probabilistic theory. The gist of our decoder is to recover the low-quality video bitstreams through a series of denoising steps. Inside each step, we try to minimize the gap between predicted noise and the true noise, which is handled by a pre-trained diffusion model. The diffusion model is established with two key concepts: (1) distortion-aware conditioning that guides model to generate high-fidelity visual details after a series of denoising steps, and (2) training the diffusion model to correctly predict the noise that should be removed to restore the frame.

Perception-aware Conditioning Similar to the conditioned generative models, we need a conditioning signal c to guide the decoder: *generating what kind of frames can minimize the fidelity gap from the original frames?* Here, the decoder should be aware of the frame distortion caused by the encoder. Therefore, we need to embed the frame distortion into conditioning signal so that the model can correctly learn the conditional probability to generate high-fidelity frames like the realistic ones. As low-quality videos are received by the ingestion server, the decoder can capture all the compressed frames and upscale them to original resolution by using the fast bilinear interpolation. Following the mainstream mechanism to handle the conditioning (Saharia

Algorithm 1: Training diffusion model $\boldsymbol{\theta}$ until convergence

Input: original frames $\mathbf{x}$**, scaling factor** s**, bit-depth** n**.**
Output: converged model $\boldsymbol{\theta}$**.**

$\hat{\mathbf{x}} = \texttt{Encode}(\mathbf{x}; s, n)$; ▷ Get the compressed frames.
repeat
  $c \leftarrow \texttt{Upscale}(\hat{\mathbf{x}})$; ▷ Fast bilinear interpolation.
  $t \sim \texttt{Uniform}(1, \ldots, T)$; ▷ Step index sampling.
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; ▷ Gaussian noise.
  Take gradient descent step on:
    $\nabla_\theta \|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t}\mathbf{x} + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}_t, t, c)\|^2$;
until $\boldsymbol{\theta}$ is converged; ▷ End with a converged model.

Algorithm 2: Inference in T steps for frame restoration

Input: compressed frames $\hat{\mathbf{x}}$**, pre-trained model** $\boldsymbol{\theta}$**.**
Output: restored frames $\tilde{\mathbf{x}}$**.**

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$;
$c \leftarrow \texttt{Upscale}(\hat{\mathbf{x}})$; ▷ Fast bilinear interpolation.
for $t \in T, \ldots, 1$ **do**
  $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; ▷ Gaussian seed.
  $\Delta\mathbf{x}_0 \leftarrow \sqrt{\bar{\alpha}_{t-1}}\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_\theta(\mathbf{x_t}, t, c)}{\sqrt{\alpha_t}}$;
  $\Delta\mathbf{x}_t \leftarrow \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, c)$;
  $\sigma_t^2 \leftarrow \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\frac{1-\alpha_t}{\alpha_{t-1}}$;
  $\mathbf{x}_{t-1} \leftarrow \Delta\mathbf{x}_0 + \Delta\mathbf{x}_t + \sigma_t\mathbf{z}$;
end for ▷ End loop with $\mathbf{x}_0$.
$\tilde{\mathbf{x}} \leftarrow \mathbf{x}_{t-1}$;
Return $\tilde{\mathbf{x}}$;

et al. 2023), the decoder initializes a Gaussian noise as the generative seed and concatenates the upscaled frames with it along the channel dimension. The diffusion model takes the concatenation result to generate high-quality frames.

Model Training and Frame Restoration Based on the discussion of embedding conditioning signals, the next step is to train the diffusion model for frame restoration. The training procedure contains two stages: (1) forward diffusion and (2) reverse diffusion, with key formulations as follows.

Forward diffusion. Given the original data distribution of the frames that $\mathbf{x} \sim q(\mathbf{x})$, the forward stage aims at gradually degrading the frame quality by inserting a small amount of Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ into the frame through T steps. Based on the original frame $\mathbf{x}_0$ at the beginning, the core formulation of degraded frame $\mathbf{x}_t$ in the t-th step ($t \in [1, T]$) can be described as: $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_{t-1} + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}_{t-1}$, where the hyper-parameter $0 < \alpha_t < 1$ controls the variance of the noise inserted in each step. Therefore, the frame $\mathbf{x}_T$ will entirely lose the visual features after T steps. When $T \to +\infty$, $\mathbf{x}_T$ is equivalent to an isotropic Gaussian distribution. Accumulating all the T steps, we can obtain the final degraded frame $\mathbf{x}_T$ as: $\mathbf{x}_T = \sqrt{\bar{\alpha}_T}\mathbf{x}_0 + \sqrt{1-\bar{\alpha}_T}\boldsymbol{\epsilon}$, where $\bar{\alpha}_T = \prod_{t=1}^{T}\alpha_t$. Thus, we can formulate the final status of the forward diffusion as: $q(\mathbf{x}_T|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_T; \sqrt{\bar{\alpha}_T}\mathbf{x}_0, (1-\bar{\alpha}_T)\mathbf{I})$. Briefly, the forward diffusion will gradually insert Gaussian noise into all the frames and finally make them equivalent to an isotropic Gaussian distribution.

Reverse diffusion. As to the frame restoration, we need to reverse the forward process of the diffusion model. This procedure is the major learning objective of our DeNC, which can be formulated as: $q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) = q(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0)\frac{q(\mathbf{x}_{t-1}|\mathbf{x}_0)}{q(\mathbf{x}_t|\mathbf{x}_0)}$. Therefore, we can train the diffusion model θ to learn this probability by minimizing the gap between the predicted noise $\boldsymbol{\epsilon}_\theta$ and true noise $\boldsymbol{\epsilon}_t$ added in the t-th step during forward stage. Following this principle, we can formulate the corresponding loss function $\mathcal{L}$ as: $\mathcal{L} = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}_t}[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t, c)\|^2]$, where t is the denoising level reflected by the step index, $\mathbf{x}_t$ is the restored sample in step t, and c is the distortion-aware conditions of low-quality video frames. By feeding these variables into the diffusion model θ, we can optimize DeNC to hold sufficient restoration capacity to generate high-fidelity videos from the compressed bitstreams. In summary, the diffusion training and video restoration procedures are summarized in Algorithm 1 and Algorithm 2, respectively.

Summary. The diffusion model inside the decoder serves as an enhancement module to restored the perceptual quality by capturing encoder's resolution-color compression.

Implementation

The diffusion model inside DeNC uses the UNet backbone (Ronneberger, Fischer, and Brox 2015) with channel pruning on the feature blocks. More precisely, we use three downsampling blocks, two middle blocks and three upsampling blocks in UNet. The scaling factors are set as 2, 4 and 8 for these three kinds of blocks, respectively. The number of base feature channels is 64. To capture the time sequence information, the diffusion step index t is specified by adding the sinusoidal position embedding into each residual block. Given a maximum step number T, we control the noise variance β_t ($t \in [1, T]$) through a linear quadratic scheduler, which gradually ranges from $\beta_1 = 10^{-4}$ to $\beta_T = 0.02$. Also, we adopt the *Mean of Squared Error* (MSE) loss with Adam optimizer (Kingma and Ba 2015) and 16 batch size to train the diffusion model. The total number of training epochs is 10K and the initial learning rate is 8×10^{-5}. During the model training stage, the resolution-color compressed frames and ground-truth frames are fed into the diffusion model to optimize its restoration capacity. Note that we use the same training setting on the baselines to optimize their restoration models, thus guaranteeing a unified evaluation criterion to compare the restoration performance.

We have restricted DeNC's computational complexity to fit the video streaming environment. In the video ingestion scenarios (Yeo et al. 2022), DeNC's restoration procedure is deployed on the media server rather than the user client. The commodity hardware on media server is powerful enough to conduct the restoration operations. For example, enhancing a 10-second video only takes 780ms when using an NVIDIA 4090 GPU, *i.e.,* lower than 8.3% additional time cost is incurred. Besides, we address the inference latency bottleneck caused by massive sampling steps. We refine the *progressive distillation* (Salimans and Ho 2022) to accelerate the inference process, where the number of required sampling steps is reduced to smaller than 10. This optimization sig-

nificantly improves the restoration speed and matches the requirements of real-time video streaming.

Evaluation

Experimental Setups

Baselines. We choose five state-of-the-art neural codecs, Gemino (Sivaraman et al. 2024), Grace (Cheng et al. 2024), LiFteR (Chen et al. 2024), NeuroScaler (Yeo et al. 2022) and NEMO (Yeo et al. 2020) as the baselines for performance comparison. To guarantee comparison fairness, all restoration models used in baselines and DeNC are trained under the same videos collected from YouTube and Netflix.

Metrics. As to the encoder's compression efficiency, we inspect how DeNC and the baselines improve the compression ratio, which is calculated based on the default H.265 (H.265 2025). Meanwhile, to evaluate neural decoder's restoration performance, we cover eight typical metrics, including PSNR (Horé and Ziou 2010), SSIM (Wang et al. 2004), VMAF (Netflix 2025b), FID (Heusel et al. 2017), UIQI (Wang and Bovik 2002), LPIPS (Zhang et al. 2018), IS (Salimans et al. 2016) and mean opinion score (MOS). They are modern quality assessment metrics used by latest neural codec methods (Saharia et al. 2023; Rombach et al. 2022; Song, Meng, and Ermon 2021; Song et al. 2021; Lugmayr et al. 2022). We also measure the per-frame time cost to inspect the computational overhead. These metrics provide a holistic comparison of the compression-restoration-overhead trade-off between DeNC and the baselines.

End-to-end Performance

Inspection of bitrate saving. Saving required streaming bitrates is the first objective of our DeNC. This is directly reflected by the video compression ratio (CR), which is calculated based on the video sizes obtained by the default H.265. The formulation is:

$$\text{CR} = \frac{\text{H.265's video size}}{\text{Other method's video size}}. \quad (1)$$

As shown in Table 1, we inspect the compression ratios achieved by DeNC and other baselines under different video settings. The resolution scaling factor is from $2\times$ to $4\times$, which are suitable to downscale common 1080p/60fps and 720p/30fps videos. The original video frames are represented in 24-bit color space and DeNC reduces the color bit-depth within $[4, 8]$. It is clear that DeNC holds much higher compression ratios over the baselines, nearly an order of magnitude. This makes DeNC qualified to stream high-definition videos in real-world scenarios.

Inspection of restoration quality. Apart from bitrate saving, providing high restoration quality is another key objective. As shown in Table 1, DeNC significantly outperforms the baselines in almost all cases, even though DeNC's compression ratios are much higher. Here, the FID is a typical metric to compare the distribution between restored frames and the original ones, providing more precise measurement over the earlier IS score. Meanwhile, SSIM is a widely-adopted metric to reflect the perceptual similarity between two frames. The highest scores of FID, IS and SSIM

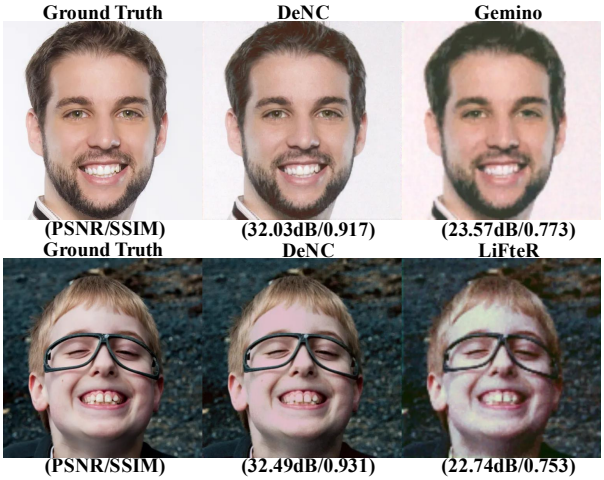


Figure 4: Comparison of restoration quality between our DeNC, Genimo and NeuroScaler. **Zoom in for best view.**

achieved by DeNC guarantee the high restoration quality for human vision, which is visualized by the case in Figure 4. Note that DeNC does not always achieve the highest PSNR because the generative procedure inside DeNC's diffusion model inserts a series of Gaussian noise to recover the visual details, which enlarges the L2 distance in PSNR. Previous work has observed that PSNR does not always correlate well with human perception, especially with a low resolution and large scaling factor (Saharia et al. 2023). This is because PSNR tends to penalize synthetic high-frequency details that are not aligned with target frames. Therefore, UIQI and LPIPS are contained to reflect human perceptual experience. We also employ the Netflix VMAF and MOS analysis (ranging from 1 to 5) to measure the coherence and perceptual quality of the restored videos, respectively.

Inspection of computational efficiency. We also inspect the parameter number and runtime cost in Table 1. Considering the large parameter scale of the vanilla UNet backbone, we conduct channel pruning on the feature blocks and restrict the parameter number with 36.2M, which fits the hardware budget of media servers (Saharia et al. 2023). As to the per-frame time cost, although DeNC is not the lowest due to the iterative sampling process of diffusion models, we optimize it by refining the progressive distillation, *i.e.,* reducing the required iteration of reverse diffusion. Thus, usually 4-8 steps will be enough. This effectively reduces the average time cost on a single frame as 21.6 ms and 42.9 ms, respectively, for 720p and 1080p videos. Since the streaming latency depends on the time cost of decoding the first video chunk, our decoding latency is about 925 ± 211 ms, which matches real-time demands in current research (Du et al. 2022; Liu et al. 2021). Currently, DeNC is not designed for cases with 4K resolution or HDR color space. However, it opens up the opportunity to unleash the potential of NeVS paradigm. A promising way for these cases is to select anchor frame referencing and adopt semantic streaming, which are open problems for further research.

Video	Method	Time (ms)	CR	PSNR↑	SSIM↑	VMAF↑	FID↓	UIQI↑	LPIPS ↓	IS↓	MOS↑
	NEMO	8.9	5.3×	32.07	0.897	92.37	4.57	0.951	0.240	1.24	2.4
720p	NeuroScaler	9.7	5.5×	32.74	0.806	93.11	4.35	0.955	0.186	1.23	2.7
30 fps	LiFteR	17.3	2.1×	33.38	0.845	94.11	3.91	0.967	0.174	1.21	3.1
YouTube	Grace	30.7	1.7×	33.77	0.828	94.41	3.83	0.971	0.167	1.19	3.3
	Gemino	97.7	9.2×	34.74	0.906	95.11	1.35	0.985	0.106	1.13	3.4
	DeNC (Ours)	21.6	31.8×	34.49	0.895	95.63	3.17	0.983	0.104	1.12	4.1
	NEMO	14.7	6.2×	27.74	0.711	91.35	8.81	0.882	0.247	2.42	2.1
	NeuroScaler	16.8	6.9×	28.62	0.728	91.58	6.77	0.903	0.221	2.25	2.2
1080p	LiFteR	32.2	1.6×	28.86	0.768	92.14	5.49	0.929	0.181	2.05	2.9
60 fps	Grace	66.8	1.3×	29.24	0.771	92.82	3.88	0.944	0.179	1.93	2.9
Netflix	Gemino	156.8	15.9×	30.77	0.782	93.16	2.76	0.952	0.097	1.85	3.1
	DeNC (Ours)	42.9	26.6×	30.68	0.771	93.41	3.19	0.953	0.064	1.28	3.9

Table 1: Performance comparison of the per-frame time cost, compression ratio (CR) and restoration quality in eight quality metrics. The video size encoded by H.265 is the base to calculate compression ratios of our DeNC and the baselines. Note that DeNC achieves a much higher compression-restoration-overhead trade-off over these latest baselines.

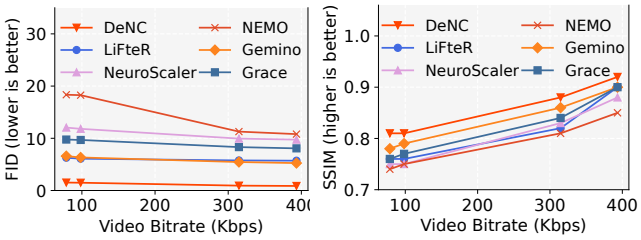


Figure 5: Comparison of the restoration quality to bitrates, where DeNC consistently outperforms existing methods.

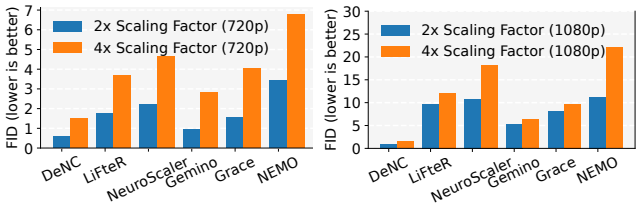


Figure 6: Restoration quality with different resolutions.

Ablation Studies

Overall compression-restoration trade-off. We inspect the trade-off between streaming bitrates and restoration quality, when using DeNC and the baselines. The bitrates are adjusted by changing the resolution scaling factor and color bit-depth. The baseline comparison can be best understood by checking Figure 5, which reports the restoration scores under different video bitrates. We can observe that a higher bitrate brings a better restoration quality, where DeNC significantly outperforms other baselines with higher trade-off. This comparison highlights DeNC's superiority in the overall compression-restoration trade-off of the NeVS pipeline.

Effect of resolution downscaling factor. We inspect how the downscaling factor impacts DeNC's restoration capability. A larger factor will compress more visual information on original frames and make the restoration procedure harder, thus leading to a worse FID score. As shown in Figure 6, our DeNC outperforms the baselines with much better FID scores under different scaling factors.

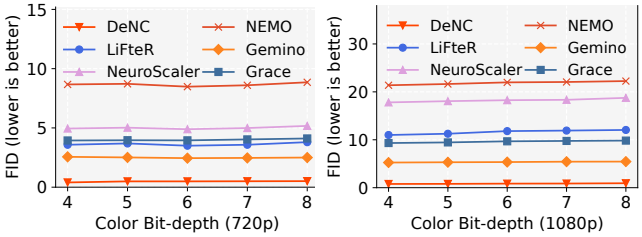


Figure 7: Restoration quality with different color bit-depth.

Effect of color bit-depth. Apart from downscaling factors, we also compare DeNC with the baselines under different bit-depths, ranging from 4 to 8. A lower bit-depth will shrink the color space and lose more distribution information of pixel values, thus also yielding a harder restoration task. Comparison results in Figure 7 show that our DeNC consistently achieves the best scores, verifying its powerful restoration capability to match human perception, even in extremely low color space.

Conclusion

The *Neural-enhanced Video Streaming* (NeVS) has been an important infrastructure for modern Internet services. We identify the performance bottleneck of current NeVS paradigm and develop new insights to unleash its potential. Aiming at improving NeVS's overall *rate-distortion-perception* trade-off, we conduct an encoder-decoder synergy and propose the *Diffusion-enhanced Neural Codec* (DeNC), a plug-and-play module to effectively reduce the streaming bitrates while guaranteeing high perceptual quality on the restored videos. Evaluations based on realistic video streaming data verify that DeNC significantly outperforms current leading methods in different quality assessment metrics, and well balances the compression ratio, restoration quality and computational overhead.

Acknowledgments

This work has been partially supported by National Natural Science Foundation of China under Grant No. U23B2026, No. 62372305 and No. 62202309, Guangdong Basic and Applied Basic Research Foundation under Grant No. 2024B1515040012, Research Team Cultivation Program of Shenzhen University under Grant No. 2023QNT015, Research Start-up Program of Shenzhen University under Grant No. RC20240254, Shenzhen Science and Technology Program under Grant No. RCBS20231211090523043, Hong Kong RGC General Research Fund (No. 15221123 and 15216424), PolyU Internal Fund (No. P0043932), and Hong Kong Generative AI Research and Development Center from InnoHK.

References

AOMedia. 2025. AOMedia Video 1 Official Website. https://aomedia.org/av1/. Accessed: 2025-01-20.

Cao, H.; Tan, C.; Gao, Z.; Xu, Y.; Chen, G.; Heng, P.; and Li, S. Z. 2024. A Survey on Generative Diffusion Models. *IEEE Trans. Knowl. Data Eng.*, 36(7): 2814–2830.

Chan, K. C. K.; Wang, X.; Yu, K.; Dong, C.; and Loy, C. C. 2021. BasicVSR: The Search for Essential Components in Video Super-Resolution and Beyond. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4947–4956.

Chan, K. C. K.; Zhou, S.; Xu, X.; and Loy, C. C. 2022a. BasicVSR++: Improving Video Super-Resolution with Enhanced Propagation and Alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5962–5971.

Chan, K. C. K.; Zhou, S.; Xu, X.; and Loy, C. C. 2022b. Investigating Tradeoffs in Real-World Video Super-Resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5952–5961.

Chen, B.; Yan, Z.; Zhang, Y.; Yang, Z.; and Nahrstedt, K. 2024. LiFteR: Unleash Learned Codecs in Video Streaming with Loose Frame Referencing. In *Proceedings of the 21st USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 533–548.

Cheng, Y.; Zhang, Z.; Li, H.; Arapin, A.; Zhang, Y.; Zhang, Q.; Liu, Y.; Du, K.; Zhang, X.; Yan, F. Y.; Mazumdar, A.; Feamster, N.; and Jiang, J. 2024. GRACE: Loss-Resilient Real-Time Video through Neural Codecs. In *Proceedings of the 21st USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 509–531.

Chu, M.; Xie, Y.; Mayer, J.; Leal-Taixé, L.; and Thuerey, N. 2020. Learning temporal coherence via self-supervision for GAN-based video generation. *ACM Trans. Graph.*, 39(4): 75.

Dasari, M.; Kahatapitiya, K.; Das, S. R.; Balasubramanian, A.; and Samaras, D. 2022. Swift: Adaptive Video Streaming with Layered Neural Codecs. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 103–118.

Du, K.; Zhang, Q.; Arapin, A.; Wang, H.; Xia, Z.; and Jiang, J. 2022. AccMPEG: Optimizing Video Encoding for Accurate Video Analytics. In *Proceedings of the Machine Learning and Systems (MLSys)*.

Google. 2025. Google VP9 Overview. https://developers.google.com/media/vp9. Accessed: 2025-01-20.

Gray, R. M.; Linder, T.; and Gill, J. T. 2008. Lagrangian Vector Quantization With Combined Entropy and Codebook Size Constraints. *IEEE Trans. Inf. Theory*, 54(5): 2220–2242.

H.265. 2025. H.265 Official Website. https://www.itu.int/rec/T-REC-H.265. Accessed: 2025-01-20.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 6626–6637.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*.

Horé, A.; and Ziou, D. 2010. Image Quality Metrics: PSNR vs. SSIM. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2366–2369.

Hu, Z.; Lu, G.; and Xu, D. 2021. FVC: A New Framework Towards Deep Video Compression in Feature Space. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1502–1511.

Kim, J.; Jung, Y.; Yeo, H.; Ye, J.; and Han, D. 2020. Neural-Enhanced Live Streaming: Improving Live Video Ingest via Online Learning. In *Proceedings of the ACM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 107–125.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Lee, R.; Venieris, S. I.; and Lane, N. D. 2022. Deep Neural Network-based Enhancement for Image and Video Streaming Systems: A Survey and Future Directions. *ACM Comput. Surv.*, 54(8): 169:1–169:30.

Li, J.; Li, B.; and Lu, Y. 2021. Deep Contextual Video Compression. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 18114–18125.

Liu, D.; Li, Y.; Lin, J.; Li, H.; and Wu, F. 2020. Deep Learning-Based Video Coding: A Review and a Case Study. *ACM Comput. Surv.*, 53(1): 11:1–11:35.

Liu, J.; Liu, P.; Su, Y.; Jing, P.; and Yang, X. 2019. Spatiotemporal Symmetric Convolutional Neural Network for Video Bit-Depth Enhancement. *IEEE Trans. Multim.*, 21(9): 2397–2406.

Liu, J.; Lu, M.; Chen, K.; Li, X.; Wang, S.; Wang, Z.; Wu, E.; Chen, Y.; Zhang, C.; and Wu, M. 2021. Overfitting the Data: Compact Neural Video Delivery via Content-aware Feature Modulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 4611–4620.

Liu, J.; Yang, Z.; Su, Y.; and Yang, X. 2022. TANet: Target Attention Network for Video Bit-Depth Enhancement. *IEEE Trans. Multim.*, 24: 4212–4223.

Lugmayr, A.; Danelljan, M.; Romero, A.; Yu, F.; Timofte, R.; and Gool, L. V. 2022. RePaint: Inpainting using Denoising Diffusion Probabilistic Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 11451–11461.

Menon, S.; Damian, A.; Hu, S.; Ravi, N.; and Rudin, C. 2020. PULSE: Self-Supervised Photo Upsampling via Latent Space Exploration of Generative Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2434–2442.

Narayanan, A.; Zhang, X.; Zhu, R.; Hassan, A.; Jin, S.; Zhu, X.; Zhang, X.; Rybkin, D.; Yang, Z.; Mao, Z. M.; Qian, F.; and Zhang, Z. 2021. A variegated look at 5G in the wild: performance, power, and QoE implications. In *Proceedings of the ACM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 610–625.

Netflix. 2025a. Netflix Official Website. https://www.netflix.com/. Accessed: 2025-01-20.

Netflix. 2025b. Video Multi-Method Assessment Fusion. https://github.com/Netflix/vmaf. Accessed: 2025-01-20.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10674–10685.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Proceedings of the Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351, 234–241.

Saharia, C.; Ho, J.; Chan, W.; Salimans, T.; Fleet, D. J.; and Norouzi, M. 2023. Image Super-Resolution via Iterative Refinement. *IEEE TPAMI.*, 45(4): 4713–4726.

Salimans, T.; Goodfellow, I. J.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved Techniques for Training GANs. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2226–2234.

Salimans, T.; and Ho, J. 2022. Progressive Distillation for Fast Sampling of Diffusion Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Shi, W.; Li, Q.; Yu, Q.; Wang, F.; Shen, G.; Jiang, Y.; Xu, Y.; Ma, L.; and Muntean, G.-M. 2024. A Survey on Intelligent Solutions for Increased Video Delivery Quality in Cloud-Edge-End Networks. *IEEE Communications Surveys and Tutorials*, 1(1): 1–1.

Sivaraman, V.; Karimi, P.; Venkatapathy, V.; Shirkoohi, M. K.; Fouladi, S.; Alizadeh, M.; Durand, F.; and Sze, V. 2024. Gemino: Practical and Robust Neural Compression for Video Conferencing. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 569–590.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Song, Y.; Sohl-Dickstein, J.; Kingma, D. P.; Kumar, A.; Ermon, S.; and Poole, B. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Wang, B.; Xu, M.; Ren, F.; Zhou, C.; and Wu, J. 2022. Cratus: A Lightweight and Robust Approach for Mobile Live Streaming. *IEEE Trans. Mob. Comput.*, 21(8): 2761–2775.

Wang, X.; Xie, L.; Dong, C.; and Shan, Y. 2021. Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 1905–1914.

Wang, Z.; and Bovik, A. 2002. A universal image quality index. *IEEE Signal Processing Letters*, 9(3): 81–84.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.*, 13(4): 600–612.

Xu, T.; Zhu, Z.; He, D.; Li, Y.; Guo, L.; Wang, Y.; Wang, Z.; Qin, H.; Wang, Y.; Liu, J.; and Zhang, Y. 2024. Idempotence and Perceptual Image Compression. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Yang, X.; Lin, H.; Li, Z.; Qian, F.; Li, X.; He, Z.; Wu, X.; Wang, X.; Liu, Y.; Liao, Z.; Hu, D.; and Xu, T. 2022. Mobile access bandwidth in practice: measurement, analysis, and implications. In *Proceedings of the ACM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 114–128.

Yeo, H.; Chong, C. J.; Jung, Y.; Ye, J.; and Han, D. 2020. NEMO: enabling neural-enhanced video streaming on commodity mobile devices. In *Proceedings of the Annual International Conference on Mobile Computing and Networking (MobiCom)*, 28:1–28:14.

Yeo, H.; Jung, Y.; Kim, J.; Shin, J.; and Han, D. 2018. Neural Adaptive Content-aware Internet Video Delivery. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 645–661.

Yeo, H.; Lim, H.; Kim, J.; Jung, Y.; Ye, J.; and Han, D. 2022. NeuroScaler: neural video enhancement at scale. In *Proceedings of the ACM International Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM)*, 795–811.

Zhang, A.; Wang, C.; Han, B.; and Qian, F. 2022. YuZu: Neural-Enhanced Volumetric Video Streaming. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 137–154.

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O. 2018. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 586–595.